

IMPLEMENTATION OF AN ANALYTICAL MODEL OF COMPRESSOR OF SMALL JET ENGINE TO ITS CONTROL SYSTEM

Norbert RÍZ, Ladislav FÓZÓ

Letecká fakulta, Technická univerzita v Košiciach
Rampová 7, 041 21 Košice
Slovenská republika
norbert.riz@student.tuke.sk, ladislav.fozo@tuke.sk

Summary. The thesis is focused on the transformation of existing mathematical model for calculating the parameters of the engine compressor ISTC-21V, which will serve to manage and diagnose engine in real time. The engine is located in the laboratory of intelligent control systems for aircraft engines in areas Faculty of Aeronautics. Existing mathematical model is created using Matlab, but motor control is carried out at laboratories in Simulink. Start of work is devoted to describing the object management software tool used Matlab / Simulink, while this chapter is an analysis of the issue. Other chapters deal with specific transformation, modifying and implementing existing matematickéh. The content of the thesis is a detailed description of the type of transformation in Matlab Simulink. The modification of an existing mathematical model required, as calculated by a compressor which is suitable for the management of the engine. The final section describes the implementation of the modified and transformed into a mathematical model of situational engine management ISTC-21V.

Keywords: Matlab, Matlab/Simulink, iSTC-21v, LIRSLM, engine control

1. INTRODUCTION

Matlab is one of the specialized software designed to solve, modeling, simulation mathematical problems. This allows you to perform complex tasks associated with modeling, controlling and diagnostics of aircraft engines. Model systems are often simulated before the system is put into operation. The mathematical model developed in Matlab is saved in a file which is called m-file. Simulink is an extension represents a software product that enables besides modeling and simulation for example, analysis of the systems that are created by using blocks.

To modeling and simulation of systems using real-time controlling pays attention currently also one of the areas of research at Department of Aerospace Engineering Faculty of Aeronautics in the laboratory of intelligent control systems for aircraft engines (LIRSLM). In the laboratory is located ISTC-21V engine, which was created from turbo-starting gear TS-20/21 and the engine MPM-20.

The main purpose of this article is creating new type of engine control, which is based on existing mathematical model used to calculate the characteristics of the examined engine. Therefore, it is imperative that m-file is modified so that instead of calculating characteristics of the engine, calculating just one point serving as the current value at the output. At the end there is a comparison of modified parameters and mathematical model of the original engine ISTC-21V.

2. REAL TIME WORKSHOP (RTW), REAL TIME WROSKHOPEMBEDDED CODER (RTW EC)

Real Time Workshop (RTW) is an extension of MATLAB-Simulink, which automatically generates code from Simulink projects for the specified Target, which represents the target hardware and software platforms for which the project generated from the Simulink resulting code [1].

Real Time Workshop Embedded Coder (EC RTW) is an extension of RTW, which is dedicated to the development of embedded devices. RTW EC allows to generate production ANSI-C, optionally ISO-C code that is highly optimized for speed, and memory requirements of the code readability. RTW EC supports code generation for the target microprocessors with fixed-point and floating-point arithmetic [1].

If we want to simulate the model in real time either use blocks of real-time libraries included in the mentioned above, or the simulation time is not entered accurately. The time parameter model will be set to infinity, ensuring continuous simulation in real time. The value of infinity is entered by typing a command `inf` instead of time value of simulation

If it is a complex project and writing each command or clicking would be time consuming and complexity increases the likelihood of incorrect entries of some variables. That's why we use editor, which allows us to save typing the command to a file called m-file used Matlab software. M-file is a text file with the extension `.m` that contains stored orders entered in Matlab, in our case it is an existing mathematical model ISTC-21V engine. M-files can be divided into two types [8]:

- Scripts - take input and do not return output arguments, operates over data Workspace
- Functions - accept input and return output arguments, internal variables are local variables functions

When the user runs the script, Matlab executes commands contained in the file line by line. Scripts can use existing data from the workspace, or they can create new data.

The scripts do not return any arguments, but the variables that are created and remain in the Workspace. Most users enter commands into the prompt Command Window which are saved in the Command History) [8].

Functions uses its own workspace, which is different from that which is used by MATLAB command line when the script run. M-file name must match the name of the function, because the function name becomes the new MATLAB statements. It is necessary to pay attention that the name does not begin with a number of other illegal characters, or did not match the name of another function in Matlab [10].

Control of ISTC-21V engine runs in Simulink, which stores the block scheme (simulation models) to a file with the extension `.mdl`. The first step of the implementation was to focus on the analysis of the linkage m-file from the MDL file. In the library of Simulink there is located S-Function block, whose great advantage for our case is that the user can define its behavior which is needed in the m-file for a MDL file. This does not represent that the existing mathematical model can be used without modification as a source code directly in the engine management. The main problem that occurs is that the existing mathematical model of the engine ISTC-21V is written as m-file script and model in Simulink block containing S-function source code used m-file function. Contents of source code except for the mathematical model will set the properties of the block S-functions.

3. IMPLEMENTATION PROCEDURE

Implementation of mathematical model for motor ISTC-21V has two main goals. The first is to use m file in Simulink, which corresponds to an existing mathematical model of compressor. The second goal is to implement the modified m-file to its control systems for engine based on LIRSLM. The implementation of mathematical model to the laboratory itself preceded some steps that are summarized in the following points:

- Transformation - the modification of an existing script m-file (the existing mathematical model of the engine ISTC-21V) to function type of m-file (source file for S-function block).

- Modification - adapting the existing mathematical model so that it can be used in situational control. The existing mathematical model calculates the characteristics of the compressor in the engine control is not required. Which means that the modification will be aimed at calculating the value of a single endpoint.
- Implementation - an activity by which a modified mathematical model will be incorporated into situational control of ISTC-21V engine to LIRSLM. Image graphics must be compatible with the Word program requirements. Figures should represent a grey-scale image. The quality of an image should be sufficient when it is readable.

3.1 Transformation

The first part of the m-file created to simulate engine parameters ISTC-21V consists of a set name. This step serves as an initialization step where you need to write the exact name of m-file as a function that will be used in the content of m-files. Every step which passes through simulation starts with the command function.

```
function Vypocet_motora (block)
setup(block);
```

To complete of any part of the m-file is not necessary to enter ending command, because if you enter the next command by function MATLAB/SIMULINK initialize method that follows the written command.

Below it is necessary to define the number of inputs, outputs and diagonal parameters, since adjustments are required for the function m-file. Input parameters in our model will be the engine speed because of rotation speed, changing the fuel supply on the basis of which it is possible to calculate the instantaneous speed. In this step, we enter only the basic parameters of S-functions therefore only numbers. If we want to enter the settings in Matlab we have to start entry setting function block.

```
function setup(block)
```

```
block.NumDialogPrms = 0;
block.NumInputPorts = 1;
block.NumOutputPorts = 3;
```

This section is further required to register using methods that will be used. On the basis that the m-file must contain outputs and the most basic registration is precisely this part, which will take the form of:

```
block.RegBlockMethod('Outputs', @Output);
```

However, besides the use of this method we have a method SetInputPortSamplingMode for that reason that we want more outputs from the user block (S-function). Register this method will be as follows:

```
block.RegBlockMethod('SetInputPortSamplingMode', @SetInputPortSamplingMode);
```

Specifying the number of inputs and outputs together from registered methods and finish the setting method and passes the specifications of each method indicated. The third part will consist of, defining the inputs and outputs:

```
function SetInputPortSamplingMode(block, idx, fd)
```

```

block.InputPort(1).SamplingMode = fd;
block.OutputPort(1).SamplingMode = 'Sample';
block.OutputPort(2).SamplingMode = 'Sample';
block.OutputPort(3).SamplingMode = 'Sample';

```

Last part will contain the output method, which will specify input and output signals. Between inputs and outputs need to enter the states. The input signal is labeled as follows:

```
n = block.InputPort (1) .dat;
```

After defining the input signal there are entered the concrete states from the existing m-file.

```

H0=0; c0=0;
kapa=1.4;
M0=c0/340;
SigmaV=0.97;
Qpl=1.2908;
%Total air pressure of the compressor
p0c=p0*(1+((kapa-1)/2)*M0^2)^(kapa/(kapa-1));
...
%Variation of parameters
deltaT_kc_pumpine = (PiKc_pumpine ^ ((kappa-1) / kappa) -1) / teta_kc_pumpine;
ddeltaT_kc_pumpine = deltaT_kc_pumpine;
deltaT_kc = (pi_kc ^ ((kappa-1) / kappa) -1) / tteta_kc;

```

At this point the created m-file which works in Simulink with our outputs. This m-file is a source code for the block S-function type Matlab Level-2.

3.2 Modification

This chapter deals with the gradual modification of the mathematical model centrifugal compressor engine ISTC-21V. Existing mathematical model of the engine is represented by m-files, which are based on knowledge surge border. Calculated basic parameters of the engine compressor ISTC-21V is plotted in characteristics such as addition $\eta_{KC} \cdot \pi_{KC} = f\left(\frac{Q\sqrt{T_{1c}}}{p_{1c}}\right)$ for different relative speeds. For the calculation of the parameters and rendering characteristics that reproduced the m-file of the dissertation [4] which was modified at the same time. In Fig. 1 is displayed, the compressor $\frac{\Delta T_{KC}}{T_{1c}} = f\left(\frac{Q\sqrt{T_{1c}}}{p_{1c}}\right)$ for different relative speeds (green color). Existing analytical model was created based on an analysis of centrifugal compressor of [3], which is calculated one course for all relative speed [2]. The expected location of the display of the results at these speeds is the fourth and sixth curve (from bottom).

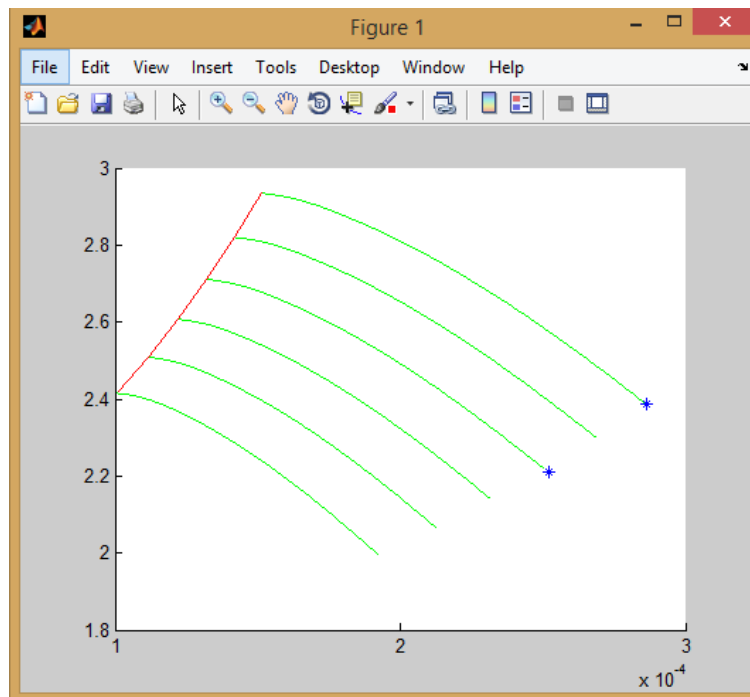


Figure 1 The result of the simulation of a modified m-file for two speeds

To characteristic of the compressor (existing m-file) were plotted calculated result values of modified m-file. The results correspond to the assumption which means that the calculated value is within the specified range but the accuracy will be verified after the formation of the model of a modified m-file. Input to S-function block is rotational speed of engine, which is entered in the Constant block. Furthermore, in the generated models there is used block Display, which shows the actual numerical value of the outgoing signal, in this case the amount of compression and efficiency of the compressor.

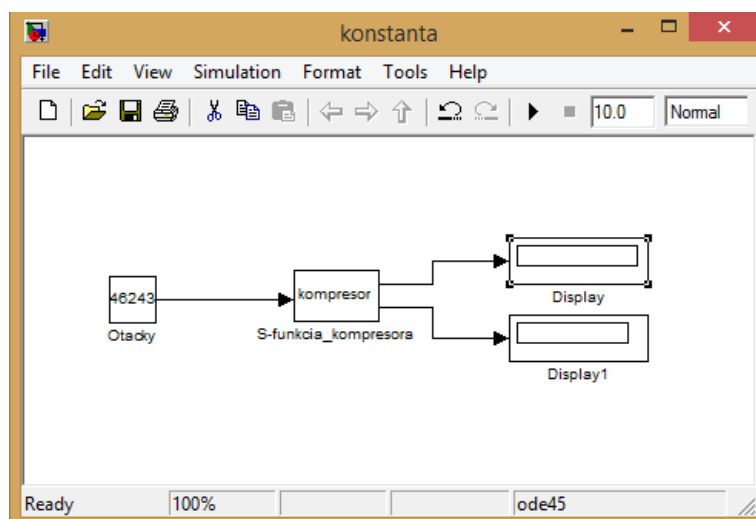


Figure 2 The simulation model in Simulink with constant input rotational speed

Described m-file that was described above is the main source of S-function block called S-funkcia_kompresora. The results of created model are compared with results of existing m-file because of correctness.

3.2.1 Testing the created m-file

This part of article consist of simulation mathematical model in real time using a gamepad in Simulink with variable rotational speed. Simulation was specified in range from 41,243 to 46,243 [rpm], because this range was specified in the original mathematical model as a range for which the calculated individual waveforms MPM20 motor parameters (current ISTC-21V). This range will form the basic range of input signals for testing and simulation model using analog of gamepad, which includes a spring. The importance of this simulation is to control the rotational speed of the current calculation, so we'll focus on the reaction speed of simulation (when the input parameter is changing). This also monitoring the accuracy of the calculation based on the previous simulation results that are also continuously checked. Speed change will be implemented using the Logitech Dual Action gamepad (Figure 3), which is supported by Simulink.



Figure 3 Gamepad - Logitech Dual Action

In the laboratory, the speed is changed by varying the angle of the joystick (which represents the Thrust lever angle - TLA), so the simulation will largely theoretical assumption given the success of the implementation of the simulation model to LIRSLM. Joystick substitution contained in LIRSLM was represented by gamepad in Fig. 3. This model was implemented in situational control but has not been tested with the engine running as it created a mathematical model needs to be complemented other parts. Joystick, which enter into LIRSLM situational model.

The involvement of gamepad to the simulation model in Simulink is necessary to use Joystick block from the library of virtual reality / 3D animation Simulink. Joystick block processes input signals from the connected game controller.

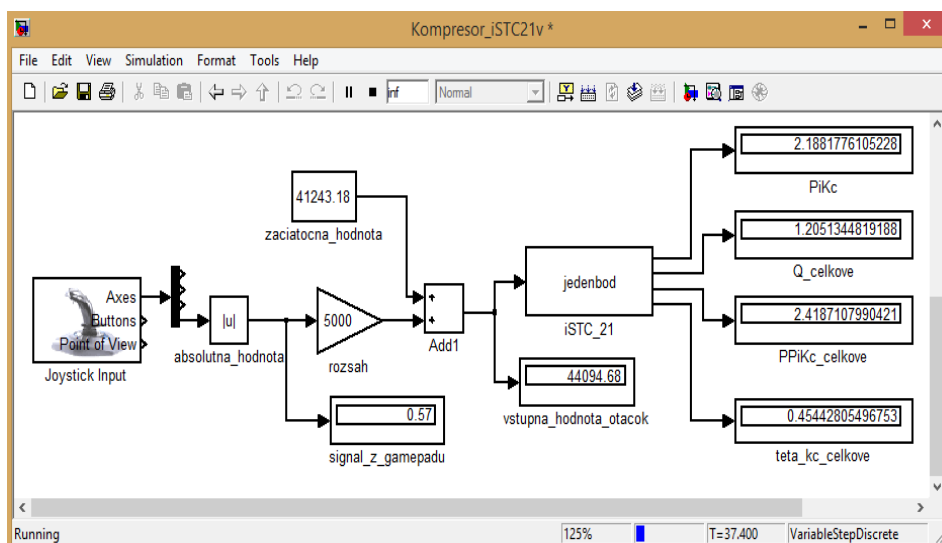


Figure 4 The simulation model with a gamepad

3.2.2 Results

Comparison parameters will consist of simulations created models with two engine speeds determined by gamepad. The results of the first simulation will correspond to an engine speed of Fig. 3. Simulations will be simulated with the 41,243,18 [rpm] of engine. The results of the simulation models will be compared with results from the Matlab workspace. The table shows the results from the original m-file labeled m-file (script).

Table 1 Results at speed $n = 44,094.68$ [rpm]

	π_{KOM}	η_{KOM}
M-file (script)	2,4187	0,4544
Block Constant	2,4187	0,4544
Block Joystick	2,4187	0,4544
Deviation_M	0	0
Deviation_M [%]	0	0

From the comparison between the original values of the m-file and simulation model with an input signal of rotational speed is clear that the modified mathematical model does not show any deviation from the original m-file.

Table 2 Comparison of results for $n=42844.68$ [rpm]

	π_{KOM}	η_{KOM}
M-file (script)	2, 3093	0,4291
Block Constant	2, 3093	0,4291
Block Joystick	2,3093	0,4291
Deviation_M	0	0
Deviation_M [%]	0	0

From the preceding tables it shows that the values before and after modifying the mathematical model are the same. Accuracy created m-file to the measured parameters according to LIRSLM [4] does not exceed 5% deviation rate. Verified parameters are accordingly acceptable deviation and it is possible to do implementation of the mathematical model of the compressor to LIRSLM. To create a complete engine control system needs a complete mathematical model examined the engine to get detailed analysis relating to the characteristics of the engine ISTC-21V.

3.3 Implementation

Location of the analytical model of the modified engine control situational ISTC-21 is shown in Fig. 5.

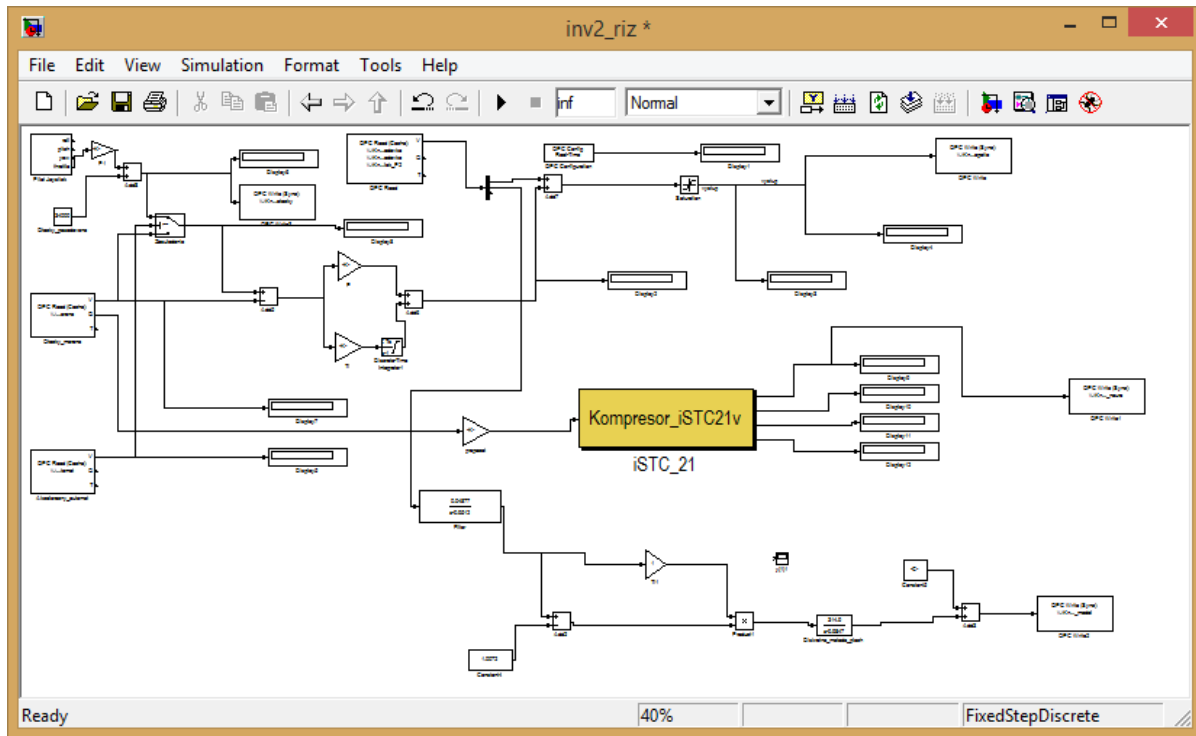


Figure 5 Implementation compressor analytic model to situational control

Despite the fact that the implementation was carried out during the writing of the thesis, the starting of engine was not realized. Next step is to extend mathematical model of on the whole engine based on these steps showed in this article. The number of input and output parameters can be easily changed as described at setting S-functions. In the future it envisages the development of a diagnostic system based on modified mathematical model. At the same time is important to create precise mathematical model of the measurement of other parameters of the engine, in order to control and diagnose the complete engine.

4 CONCLUSION

This article includes analysis capabilities and the sequence of the implementation of the mathematical model of the engine compressor ISTC-21V to situational control. The implementation itself precedes the adjustments like transformation and modification of an existing model.

We concluded that the ability to define S-function is very comprehensive and user can define any library what he wants. In this article there was described how to convert script m-file to function m-file. The created m-file can be used for engine control because it has allowed deviation. The modification of all parts of the engine ISTC-21V will continue to create entire engine control. It also has the additional tasks connected with visualization engine ISTC-21V, which would be displayed measured parameters displayed in the graphical interface MATLAB/SIMULINK. For a more comfortable reading measured data in model in Simulink blocks could be added as the visualization of measured parameters, the minimum resp. maximum values of individual parameters of the analytical model.

References

- [1] BYSTERSKÝ T: Podpora hardware periférií při generování kódu ze Simulinku. Diploma thesis; Praha, 2008, page 80
- [2] FŐZŐ L: Využitie matematického modelu rovnovážneho a nerovnovážneho chodu motora MPM20 pri návrhu algoritmu riadenia v každom čase. Dissertation thesis Košice, 2008. page 61-91
- [3] RŮŽEK, J: Teorie leteckých motorů, Část I. VA AZ Brno, 1979, page 373.
- [4] FŐZŐ, L. - ANDOGA, R. – MADARÁSZ, L. - KOLESÁR, J. - JUDIČÁK, J: Description of an intelligent small turbo-compressor engine with variable exhaust nozzle. SAMI 2015.
- [5] KOMJÁTY, M., HOCKO, M.: Application of the Method of Small Alterations to Analyse the Fundamental Parameters of a Small Turbojet Engine MPM-20, In: Acta Avionica Journal. - September 2014. - ISSN 1335-94794. - Volume XVI, 29 – No. 1. Faculty of Aeronautics, Košice, Slovakia.
- [6] Považan J.: Konštrukcia matematických modelov leteckých turbo-kompresorových motorov, VLA M.R.Š. v Košiciach, 1999, ISBN 80-7166-030-2.
- [7] Madarász, L.: Inteligentné technológie a ich aplikácie v zložitých systémoch. Vydavateľstvo Elfa, s.r.o., TU Košice, 349 pp., ISBN 80 – 89066 – 75 - 5, 2004.
- [8] Matlab – M-súbory a Editor [online] [5.5.2015]. Available on: <<http://www.posterus.sk/?p=1812>>